

REMARKS

To better contrast the prior art against the claimed invention, Applicants will first elaborate a bit about the Angelo reference (USP 5,923,754). Consider the flowchart shown in Angelo's Figure 3. As shown by elements 56 and 58, Angelo's DVD disk stores both a disk key and a media key. These are all that are needed to decode the content on the DVD. To provide extra security, the disk drive encodes these keys before they are transmitted to the video controller (see step 60 and 64). The video controller then decrypts these keys (step 66) so that the content on the DVD may be decrypted using the decrypted disk key (step 72). But look at the glaring problem: the keys to the content are stored on the DVD disk. Thus, suppose a hacker reads them using a "non-Angelo" disk reader. He/she now has the keys to the content. This was pointed out in the last response but evidently misunderstood. The point is not that Angelo encrypts the keys before transmitting them to the video controller: that is plain from the patent. Instead, the point is that the keys to the content sit on the disk!!

In the final office action, it was indicated that "[o]n the contrary, Angelo teaches that the DVD drive after querying the data key and the medium key from the disk, encrypts them and then transfers them to the system memory." Yes, Applicants agree that Angelo's disk drive does indeed encrypt the keys in this fashion. But, that is not Applicant's point: instead the point is that a hacker would not use an "Angelo" disk drive. Instead, the hacker would read the encoded content and keys using a conventional disk drive and store them, for example, in the memory of a PC. Using a software program, the hacker could then identify the keys vs. the encoded content. The hacker's task is then accomplished because once you have the keys it is a simple matter to then decrypt the encoded content with them. Thus Angelo does not provide the advantageous protection as set forth in claim 1.

As discussed, for example, in the background section of the application, prior art encryption schemes suffered from a number of disadvantages. For example, should the user

have to enter a password or code to access data stored on storage medium, use becomes cumbersome should the storage medium be transferred to a new machine as the user must remember the password and re-enter it. However, storing the decryption key on the player device or host system is also undesirable because the decryption key is then vulnerable to hacking – this, however, is exactly what is practiced in Angelo.

To solve these problems in the prior art, Applicants have provided systems and methods that do not require a user to enter a password nor is the password vulnerable to hacking. For example, consider claim 1, which recites a method having the acts of: “generating an internal key using the data storage engine; generating a combination key by combining a the medium key with the internal key; and decrypting a first portion of data stored on the storage medium with said first combination key.” For clarification, Applicants have amended claim 1 to replace “using” with “with” to more particularly point and distinctly claim the point that it is the combination key that decrypts the first portion of data. In sharp contrast, the Angelo reference discloses that a disk key stored on the disk is what is used to decrypt the content. Referring again to claim 1, the medium key is, as the name suggests, a key stored on the storage medium. However, a user cannot simply hack this medium key to unlock the content contained on the storage medium. Instead, the storage engine must generate the internal key and combine it with the medium key to produce a combination key, which may then be used to access the content. Such a method solves the problems associated with the prior art in that the combination key must be generated by the storage engine. As such, it is not stored on medium, making it vulnerable to hacking. Moreover, because it is internally generated by the storage engine, it is not accessible by the host system. These advantageous features provided by the method recited in claim 1 are neither taught nor suggested by the cited prior art. For example, consider the Angelo reference (USP 5,923,754), which is directed to an encryption scheme to protect the data transfer from a DVD

drive to a host's video controller/video monitor (see, e.g., Figure 2 depicting the DVD drive as element 12 and the video controller/monitor as elements 18 and 20). In this scheme, the DVD drive reads a "disk key" and a "media key" directly from the DVD – see, e.g., Figure 3. But note that the abstract refers to these keys as the "data key" and the "medium key." Regardless of what they are called, it is these keys that are used to decrypt the content. Thus, the keys are vulnerable to hacking as described above. To protect them, the Angelo reference discloses an encryption scheme between the DVD drive and the host video system. The DVD drive generates an internal key (referred to as the first key in the Abstract) that is combined with a key stored in the host video system (referred to as the output key in the Abstract) to form a "second secure transfer key." This second secure transfer key is then used to encrypt the data key and medium key before they are passed to the host video system, which then decrypts to recover the data key and the medium key. But note the glaring problem: a DVD drive that merely transferred the data key and the medium key without encryption would leave the content vulnerable to hacking.

In sharp contrast, the method recited in claim 1 requires the generation of a combination key to unlock the encrypted content. This combination key is NEVER stored on the storage medium as disclosed in Angelo. Instead, the storage engine must first generate an internal key and then combine the internal key with the medium key to produce the combination key, which is then used to unlock the encrypted content. As such, the method recited in claim 1 is superior to that disclosed in Angelo. Because Angelo makes no teaching or suggestion for such an advantageous method, claim 1 is patentable over the Angelo reference.

The Silverbrook reference (USP 6,334,190) adds nothing further as it is directed to authentication schemes to prevent users from, for example, using non-authenticated printer cartridges as refills in printers. As such, Silverbrook does nothing to cure the deficiencies in

the Angelo references. Accordingly, claim 1 is patentable over the combination of Angelo and Silverbrook.

Because claims 2 -3, and 6 through 13 depend either directly or indirectly upon claim 1, they are patentable over the Angelo and Silverbrook references for at least the same reasons. Claims 4 and 5 are cancelled, thereby mooted their rejections.

Claim 14 is patentable over the Angelo and Silverbrook references analogously as discussed with respect to claim 1. It has also been amended to replace "using" with "with" in the same fashion as was done in claim 1. The cited references make no teaching or suggestion for the provision of "a plurality of combination keys from the plurality of medium keys and the plurality of internal keys; and decrypting a first portion of the data with a first combination key from the plurality of combination keys." Because claims 15 through 21 depend either directly or indirectly upon claim 14, they are patentable over the art of record for at least the same reasons.

Claims 22 through 25 stand withdrawn as being directed to a non-elected species.


CONCLUSION

For the above reasons, pending Claims 1 ~ 3, and 6 - 21 are in condition for allowance and allowance of the application is hereby solicited. If the Examiner has any questions or concerns, a telephone call to the undersigned at (949) 752-7040 is welcomed and encouraged.

I hereby certify that this correspondence is facsimile transmitted to the Commissioner for Patents, Washington, D.C. 20231, at 703-872-9306, on July 27, 2004.


Jonathan HallmanJuly 27, 2004
Date of Signature

Respectfully submitted,


Jonathan W. Hallman
Attorney for Applicant(s)
Reg. No. 42,622